WHAT IS CLAIMED IS:

1. A method for recoverable programming, comprising the steps of:

identifying a predetermined instruction sequence;

monitoring for memory access errors;

logging a memory access error in an error logging register;

polling the register for any logged memory access error during execution of the instruction sequence; and

raising exceptions, if the memory access error is logged.

2. The method of claim 1, further comprising the steps of:

checkpointing a predetermined set of system data; and

recovering from the memory access error using the checkpointed system data, if the memory access error is logged during execution of the instruction sequence.

3. The method of claim 1, further comprising the step of:

setting data returned in response to the memory access request equal to a set of predefined fake data, if the memory access error is logged during execution of the instruction sequence.

4. The method of claim 3, further comprising the step of:

skipping the polling and raising steps if the data returned in response to the memory access request is not equivalent to the predefined fake data.

5. The method of claim 1, further comprising the step of:

masking a machine check abort handle.

6. The method of claim 5, after the raising step, further comprising the steps of:

enabling the machine check abort handle.

7. The method of claim 1, further comprising the step of:

updating pointers, if the memory access error is logged.

8. The method of claim 1, further comprising the step of:

re-executing the memory access request, if software so commands.

9. A method for recoverable programming, comprising the steps of:

identifying a predetermined instruction sequence;

checkpointing a predetermined set of system data;

masking a machine check abort handle;

monitoring for memory access errors;

logging a memory access error in an error logging register;

polling the register for any logged memory access error during execution of the instruction sequence;

raising exceptions, if the memory access error is logged;

updating pointers, if the memory access error is logged;

recovering from the memory access error using the checkpointed system data, if the memory access error is logged during execution of the instruction sequence.;

re-executing the memory access request, if software so commands; and enabling the machine check abort handle.

10. A computer-usable medium embodying computer program code for commanding a computer to perform recoverable programming, comprising the steps of:

identifying a predetermined instruction sequence;

monitoring for memory access errors;

logging a memory access error in an error logging register;

polling the register for any logged memory access error during execution of the instruction sequence; and

raising exceptions, if the memory access error is logged.

11. The medium of claim 10, further comprising the steps of:

checkpointing a predetermined set of system data; and

recovering from the memory access error using the checkpointed system data, if the memory access error is logged during execution of the instruction sequence..

12. The medium of claim 10, further comprising the step of:

setting data returned in response to the memory access request equal to a set of predefined fake data, if the memory access error is logged during execution of the instruction sequence.

13. The medium of claim 13, further comprising the step of:

skipping the polling and raising steps if the data returned in response to the memory access request is not equivalent to the predefined fake data.

14. The medium of claim 10, further comprising the step of:

masking a machine check abort handle.

15. A system for recoverable programming, comprising:

means for identifying a predetermined instruction sequence;

means for monitoring for memory access errors;

means for logging a memory access error in an error logging register;

means for polling the register for any logged memory access error during execution of the instruction sequence; and

means for raising exceptions, if the memory access error is logged.

16. The system of claim 15, further comprising:

means for checkpointing a predetermined set of system data; and

means for recovering from the memory access error using the checkpointed system data, if the memory access error is logged during execution of the instruction sequence..

17. The system of claim 15, further comprising:

means for setting data returned in response to the memory access request equal to a set of predefined fake data, if the memory access error is logged during execution of the instruction sequence.

18. The system of claim 17, further comprising:

means for bypassing the means for polling and means for raising if the data returned in response to the memory access request is not equivalent to the predefined fake data.

19. The system of claim 15, further comprising the step of:

means for masking a machine check abort handle.